

# Software-Agenten im Semantic Web

Rolf Grütter

**Die Entwickler des Semantic Web orientieren sich in ihrem Verständnis von Software-Agenten an der Konzeption der Agentenkommunikationssprache ACL von DARPA. Dabei werden Agenten aufgrund ihrer Fähigkeit, korrekt in ACL zu kommunizieren, definiert.**

ACL besteht aus drei Teilen: einem in einer Ontologie definierten Vokabular, einer inneren Sprache für den deklarativen Nachrichteninhalte und einer äusseren Sprache für prozedurale Anweisungen. Von den drei Teilen sind im Semantic Web die Ontologien am

weitesten entwickelt, und mit OWL ist eine standardisierte Beschreibungssprache verfügbar. Als innere Sprache eignet sich eine ausdrucksstarke Sprache wie webgängiges KIF. OWL-QL ist ein Vorschlag für eine äussere Sprache, die für den Gebrauch mit OWL spezifiziert wurde und Frage-Antwort-Dialoge sowie Schlussfolgerungen in einer Beschreibungslogik unterstützt. RuleML und die im Rahmen von REVERSE entwickelten Sprachen werden das Repertoire der von den Software-Agenten des Semantic Web unterstützten Schlussfolgerungen und ihre Interaktionen reichhaltiger und vielfältiger gestalten als OWL und OWL-QL.

## Einführung

Software-Agenten sind die Benutzerschnittstelle zum Semantic Web. Als virtuelle Handlungsreisende bevölkern sie das Semantic Web und führen für ihre menschlichen Benutzer Aufträge aus. Dazu müssen sie mit anderen Software-Agenten kommunizieren und ihre Dienste ansprechen, das heisst, mit ihnen interagieren können. Das bedeutet aber nicht, dass die Benutzer des Semantic Web nicht mehr aktiv in

die Prozesse eingebunden sind. Hendler [20] weist darauf hin, dass Software-Agenten, statt die Bedürfnisse der Benutzer unmittelbar zu stillen, vielmehr verschiedene Wege, die zu diesem Ziel führen, ausfindig machen und dem Benutzer dann die Wahl überlassen. Software-Agenten werden im Semantic Web deshalb vor allem Hilfsarbeiten erledigen, wie das erwähnte Bereitstellen einer geeigneten Entscheidungsgrundlage. Dies steht im Einklang mit der Konzeption der heutigen Suchmaschinen des *World Wide Web (WWW)*. Insofern stellt das Semantic Web – zumindest für den Benutzer – keinen Paradigmenwechsel dar, sondern eine organische Weiterentwicklung und Verbesserung.

Trotz der zentralen Bedeutung von Software-Agenten unterhält das *World Wide Web Consortium (W3C)* keine separate Aktivität in diesem Bereich. Dies mag auf den ersten Blick erstaunen. Bei genauerem Hinsehen wird aber deutlich, dass sich die Entwickler des Semantic Web an Vorarbeiten an *Software-Agenten* orientieren und das W3C mit den für das Semantic Web spezifischen Empfehlungen (*Recommendations*) gleichsam *Bedingungen für Software-Agenten* formuliert. Darüber hinaus werden in mehreren Veröffentlichungen Bemerkungen über Agenten gemacht. Ein Überblick über diese Bemerkungen und der sich darin ausdrückende Begriff

DOI 10.1007/s00287-005-0040-1  
© Springer-Verlag 2005

Institut für Medien- und Kommunikationsmanagement  
Universität St. Gallen  
Korrespondenzadresse:  
Dr. Rolf Grütter  
Föhrenstrasse 4  
CH-9000 St. Gallen  
Schweiz  
E-Mail: rolf.gruetter@unisg.ch

## Software Agents and Semantic Web

The developers of the Semantic Web use DARPA's Agent Communication Language (ACL) to establish a common understanding of software agents. The definition of agents is therefore based on their ability to communicate in ACL correctly. ACL is comprised of three parts: a vocabulary that is defined in an ontology, an inner language containing the declarative message content, and an outer language for procedural instructions. Of these three parts the ontologies are developed the furthest in the Semantic Web. With OWL there even is a standardized description language available. As an inner language expressive languages like webized KIF qualify. OWL-QL has been proposed as an outer language. It has been specified to be used with OWL and supports query-answering dialogs as well as reasoning in description logics. RuleML and the languages developed in the context of REWERSE will augment the reasoning capabilities of the Semantic Web software agents and the supported types of interaction beyond the scope of OWL and OWL-QL.

des Agenten steht am Anfang der folgenden Ausführungen. Am Schluss werden die Software-Agenten des Semantic Web anhand der gebräuchlichen Agententheorien, -architekturen und -sprachen beschrieben und Beispiele für Anwendungen von Software-Agenten gegeben.

## Abkürzungsverzeichnis

ACL	Agent Communication Language
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
DL	Description Logics
DLP	Description Logic Programs
EU	European Union
HTTP	Hypertext Transfer Protocol
IRC	Internet Relay Chat
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
KSE	Knowledge Sharing Effort

OIL	Ontology Inference Layer/Ontology Interchange Language (vornals)
OWL	Web Ontology Language
OWL-QL	OWL Query Language
RDF	Resource Description Framework
REWERSE	Reasoning on the Web with Rules and Semantics
RuleML	Rule Markup Language
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SWAD	Semantic Web Advanced Development
SWRL	Semantic Web Rule Language
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

## Zum Begriff des Agenten

Der Begriff des Agenten wird weit gefasst und schliesst alle im Semantic Web Handelnden – Rechner (-programme) und Menschen – gleichermaßen ein: „The Semantic Web is a collection of computers and people exchanging information. Collectively, we can call them agents.“ (Basics, Abschn. 1) [19]. Die Rechnerprogramme umfassen die Software-Agenten des WWW – Browser, Multimedia-Abspielgeräte, Suchmaschinen, Proxy-Server und Server [24] – und die *spezifischen* Software-Agenten des Semantic Web. Letztere spielen eine besondere Rolle [5]:

The real power of the Semantic Web will be realized when people create many programs that collect Web content from diverse sources, process the information and exchange the results with other programs. The effectiveness of such software agents will increase exponentially as more machine-readable Web content and automated services (including other agents) become available. The Semantic Web promotes this synergy: even agents that were not expressly designed to work together can transfer data among themselves when the data come with semantics (Agents, Abschn. 1).

Im selben Artikel unterscheiden Berners-Lee, Hendler und Lassila – anhand der gespielten Rollen – zwischen *Benutzeragenten* (Consumer Agents) und *Dienstagenten* (Producer Agents). Das im Englischen verwendete Wort „Konsument“ bezieht sich dabei auf den menschlichen Benutzer, der – zum Beispiel – angeforderte Informationen „verwertet“, das Wort „Produzent“ auf den Dienstleister, der die

Informationen erzeugt und anbietet. An anderer Stelle benutzt Berners-Lee den Begriff *Semantic Web Agents*, um jene Software-Agenten einzugrenzen, die auf der Grundlage der Technologien des Semantic Web zu gewissen Folgerungen oder *Inferenzen* befähigt sind [4]. Diese Sichtweise, welche die Software-Agenten des Semantic Web als Benutzer und Dienstleister versteht, widerspiegelt einen aktuellen Trend auf dem Gebiet der agentengestützten Verarbeitung, nämlich hin zur Entwicklung von Web Services [1]. (Die für Web Services spezifischen Empfehlungen des W3C und die in Entwicklung befindlichen Technologien für *Semantic Web Services* werden in dieser Arbeit nicht berücksichtigt. Der Artikel befasst sich schwergewichtig mit den Benutzeragenten im Sinne der oben eingeführten Unterscheidung.)

Auch zum Inhalt des Agentenbegriffes macht das W3C nur wenige Angaben. Im oben zitierten Abschnitt sind Software-Agenten in der Lage, Informationen aus verschiedenen Quellen zu sammeln, zu verarbeiten und die Ergebnisse mit anderen Agenten auszutauschen. Hawke macht deutlich, dass Agenten Wissen haben und Handlungen ausführen [19]. Dabei unterstellt er einen Zusammenhang zwischen Wissen und Handlungen: „Agent’s actions are assumed to be based entirely on their knowledge ...“ (Basics, Abschn. 1). Die Handlungssteuerung erfolgt damit ausschliesslich mit dem Wissen, das ein Agent erwirbt oder erschliesst. Dazu gehört auch das Wissen über die Bedürfnisse und Fähigkeiten von anderen Agenten sowie Fragen als wohldefinierte Wissenslücken und Ziele als Wissen über erwünschte zukünftige Zustände der Welt (Fragen sind eine besondere, als *Information Gathering Goals* bezeichnete Art von Zielen, und ihre Erreichung verändert – im Gegensatz zur Erreichung der als *Achievement Goals* bezeichneten Ziele – nur den Zustand der Welt *des Agenten*, nicht aber den allgemeinen Zustand der Welt [33]).

## Software-Agenten

### Eine prototypische Agentenkommunikationssprache

Die Konzeption des Semantic Web, dass Agenten durch den Austausch von *semantisch beschriebenen* Daten zur Zusammenarbeit befähigt werden, hat ein Vorbild in der im Rahmen von DARPA KSE entwickelten *Agent Communication Language (ACL)* [16].

Anders als die Nachrichten in objektorientierten Programmen ist die Semantik dieser Sprache von den „Objekten“ – den Agenten – unabhängig. Agenten werden aufgrund ihrer Fähigkeit, korrekt in einer Sprache wie ACL zu kommunizieren, definiert: „The criterion for agenthood is a behavioral one. An entity is a software agent if and only if it communicates correctly in an agent communication language such as ACL“ (S. 50) [16]. Das heisst, dass ein Software-Agent in der Lage sein muss, Nachrichten in ACL zu lesen und zu schreiben, aber auch, dass er sich an die von den Inhalten dieser Nachrichten vorausgesetzten Verhaltensregeln halten muss.

Die zu einer Nachricht gehörigen Verhaltensregeln leiten sich daneben auch aus allgemeinen Verhaltensprinzipien für Agenten her [16]. Solche Verhaltensprinzipien sind zum Beispiel *Aufrichtigkeit (Veracity)*, *Autonomie (Autonomy)* und *Verpflichtung (Commitment)*. *Aufrichtigkeit* meint, dass ein Agent die Wahrheit erzählen muss bzw. dass er nur solche Verpflichtungen eingeht, von denen er glaubt, sie erfüllen zu können [39]. *Autonomie* meint, dass ein Agent den anderen zur Leistung eines Dienstes nicht zwingen darf, es sei denn, der andere Agent habe seine Bereitschaft, Dienst Anfragen anzunehmen, zuvor angezeigt. *Verpflichtung* meint, dass ein Agent einen Dienst auf Anfrage hin leisten muss, wenn er seine Bereitschaft dazu angezeigt hat. Shoham erwähnt zusätzlich, dass die Überzeugungen eines Agenten in sich *konsistent* sein müssen [39], was auch für die Wissensbasen der Software-Agenten des Semantic Web zutrifft (s. *Semantik der möglichen Welten* im Abschnitt *Beschreibung der Software-Agenten*), nicht aber für das Semantic Web als Ganzes!

ACL besteht – wie in Abb. 1 gezeigt – aus drei Teilen: einem Vokabular, einer inneren Sprache, *Knowledge Interchange Format (KIF)*, und einer äusseren Sprache, *Knowledge Query and Manipulation*

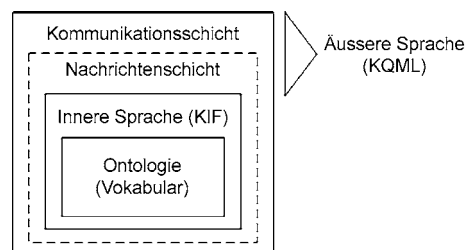
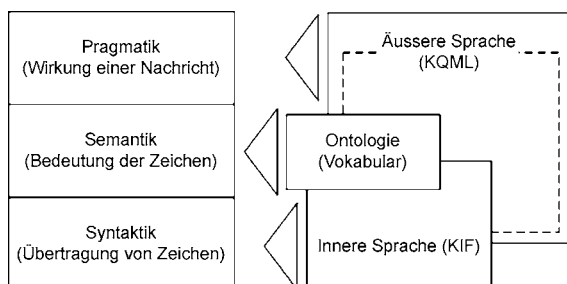


Abb. 1 Agent Communication Language (ACL)

*Language (KQML)*. Eine Nachricht in ACL ist ein Ausdruck in KQML, dessen „Argumente“ mit Wörtern aus dem Vokabular gebildete Ausdrücke oder Sätze in KIF sind [16].

Das für den Austausch von Nachrichten in einem spezifischen Anwendungskontext verwendete Vokabular entstammt einer für die Beschreibung des Kontextes geeigneten Ontologie mit den formalen Definitionen der Termini. KIF ist eine kompakte Version von Prädikatenlogik erster Stufe mit verschiedenen Erweiterungen, welche seine Ausdruckskraft erhöhen [17]. KQML bietet für die Ausdrücke in KIF eine zusätzliche Schicht, in welcher durch das Einbinden von Informationen über den Sender, den Empfänger, die Zeit usw. der Nachrichtenkontext berücksichtigt wird.

Die mit KQML zusätzlich angebotene Schicht wird weiter in eine Kommunikationsschicht und eine Nachrichtenschicht unterteilt [12–14]. Die *Kommunikationsschicht* kodiert einen Satz von Eigenschaften, welche die grundlegenden Parameter der Kommunikation beschreiben, wie die Identität des Senders und Empfängers. Die *Nachrichtenschicht* bestimmt die Art der Interaktion. Mit einem *performativen Verb (Performative)* – zum Beispiel *ask-one* im Falle einer an genau einen anderen Agenten gerichteten Frage – wird der illokutive Sprechakt und damit zugleich das Interaktionsprotokoll festgelegt. Der *illokutive* Akt bezeichnet jenen Aspekt der mit einer Äusserung vollzogenen Sprachhandlung, mit welchem im Hörer eine bestimmte, als *perlokutiver* Akt bezeichnete, Wirkung erzielt werden soll [38]. Zum Beispiel will der Sprecher mit Fragen (illokutiver Akt) den Hörer zum Antworten bewegen (perlokutiver Akt). Weitere – optionale – Eigenschaften beschreiben zum Beispiel die Sprache des Nachrichteninhalts (d. h. die innere Sprache) und die vorausgesetzte Ontologie.



**Abb. 2 Ebenen der Kommunikation und ACL**

Die drei Schichten von ACL lassen sich – wie in Abb. 2 gezeigt – den aus der allgemeinen Sprachtheorie stammenden drei Ebenen der Kommunikation zuordnen. So entspricht die innere Sprache (z. B. KIF) der syntaktischen Ebene, die Ontologie, welcher die von der inneren Sprache verwendeten Wörter entstammen, der semantischen Ebene und die äussere Sprache (z. B. KQML) mit den performativen Verben der pragmatischen Ebene.

Die Ebene der Syntaktik befasst sich dabei mit der korrekten *Übertragung der Zeichen* und mit ihrer formalen Beziehung zueinander [34]. Auf der Ebene der Semantik stehen die *Bedeutung der Zeichen* und die Inhalte der Nachrichten im Vordergrund. Die Ebene der Pragmatik befasst sich schliesslich mit der *Wirkung*, die der Sender mit seiner Nachricht beim Empfänger erzielen möchte.

## Eine Agentenkommunikationssprache für das Semantic Web

Im Zuge des Ausbaus des WWW wurden sowohl für KIF als auch für KQML auf das WWW portierbare, als *webized* bezeichnete Sprachversionen vorgeschlagen, welche mögliche Ausgangspunkte für eine Agentenkommunikationssprache im Semantic Web darstellen. Von einer solchen Sprachversion wird im Wesentlichen verlangt, dass die Namen, Kürzel und Bezeichner durch URIs ersetzt werden [3]. Um eine leichte Einbindbarkeit in die vorhandenen Technologien des WWW und des Semantic Web zu gewährleisten, ist im Weiteren eine konkrete Syntax in XML oder in einer XML-Anwendung wie RDF erforderlich. Als Beispiel sei die Reformulierung von KIF als RDF Schema erwähnt [8].

Eine strikte Trennung des deklarativen Inhalts vom prozeduralen performativen Verb, welche ACL und insbesondere KQML auszeichnet, wird auch bei der Entwicklung der für das Semantic Web vorgesehenen *Rule Markup Language (RuleML)* angestrebt [7]. Die mit RuleML ergriffene Initiative wird im Rahmen des von der EU unterstützten *Network of Excellence REVERSE (Reasoning on the Web with Rules and Semantics)* weiterverfolgt und mit Blick auf die Verbindung mit anderen Sprachen zur Unterstützung von Schlussfolgerungen (*Reasoning Languages*) ausgeweitet [35]. Einer ähnlichen Konzeption wie RuleML folgt OWL-QL – eine prototypisch ausgelegte, deduktive Abfragesprache für das Semantic Web.



OWL Query Language (OWL-QL) ist eine für die Weiterentwicklung zum Standard vorgeschlagene Sprache und ein Protokoll für Frage-Antwort-Dialoge zwischen Software-Agenten, die von in OWL repräsentiertem Wissen Gebrauch machen [11]. OWL-QL legt die semantischen Beziehungen zwischen einer Frage, der Antwort auf diese Frage und der zum Erzeugen der Antwort verwendeten Wissensbasis (oder Wissensbasen) genau fest. Anders als die gängigen Abfragesprachen von Datenbanken unterstützt OWL-QL Dialoge, bei denen der antwortende Agent eine Antwort mit den Methoden der automatisierten Beweisführung (*Reasoning*) herleitet ebenso, wie Dialoge bei denen das zur Beantwortung der Frage verwendete Wissen in mehreren Wissensbasen im Semantic Web liegt und diese Wissensbasen durch den fragenden Agenten nicht angegeben werden (im Gegensatz zu SQL geht OWL-QL von der Voraussetzung aus, dass nicht triviale Folgerungen vom *antwortenden* Agenten vollzogen werden und nicht vom fragenden). Obwohl OWL-QL für den Gebrauch mit OWL spezifiziert wurde, ist die Sprache so ausgelegt, dass sie leicht an andere deklarative Wissensrepräsentationssprachen wie KIF, RDF und DAML+OIL angepasst werden kann. OWL-QL ist für das Semantic Web ausgelegt: Die Fragen und Antworten in XML-Syntax werden zur Übermittlung mit einer Umhüllung durch das *Simple Object Access Protocol (SOAP)* versehen [29], und ein Teil der pragmatischen Aspekte der Kommunikation wird auf den Ebenen von HTTP und der Protokolle des Internets geregelt.

### Architektur eines Mehragentensystems

Genesereth und Ketchpel schlagen vor, die Agenten als *föderiertes System* zu organisieren [16]. In einem föderierten System kommunizieren die Agenten – wie in Abb. 3 gezeigt – nicht direkt miteinander,

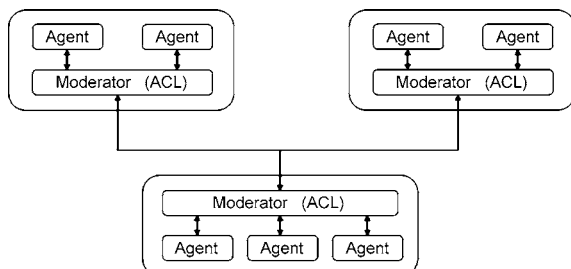


Abb. 3 Architektur eines Mehragentensystems (adaptiert nach [40])

sondern unter Vermittlung von als *Moderatoren (Facilitators)* bezeichneten Systemprogrammen. Der Vorteil dieser Architektur gegenüber der direkten Kommunikation ist, dass weniger Nachrichten ausgetauscht werden müssen und die Komplexität der Implementierung der Agenten geringer ist, weil ein Agent nur mit dem lokalen Moderator und nicht mit anderen Agenten verhandeln muss.

In einem föderierten System benutzen die Agenten ACL, um ihre Bedürfnisse und Fähigkeiten zu dokumentieren, aber auch um mit Hilfe von Nachrichten Informationen anzufordern und zu liefern [16]. Die Moderatoren benutzen die von den Agenten zur Verfügung gestellten Dokumentationen, um die Nachrichten an die zuständigen Stellen weiterzuleiten. Bei Bedarf können die Moderatoren die Nachrichten von einem Vokabular ins andere übersetzen, indem sie die Definitionen aus den verwendeten Ontologien abrufen. Eine Unterstützung kann zum einen dadurch geleistet werden, dass der Nachrichtenfluss über die Moderatoren geleitet wird. Zum andern stellen die Moderatoren besondere Verbindungen zwischen einzelnen Agenten her und treten dann in den Hintergrund.

Auch wenn die von Genesereth und Ketchpel [16] vorgeschlagene zweistufige Architektur wegen der grossen Zahl der angeschlossenen Systeme nicht direkt auf das Semantic Web übertragbar ist, sind es die zugrunde liegenden Konzepte. So werden auch im Semantic Web spezialisierte Agenten Auskunft darüber geben, welche anderen Agenten eine gegebene Anfrage bearbeiten können. Ebenso werden spezialisierte Agenten von einem gegebenen Vokabular in ein anderes übersetzen.

Das erste Konzept findet im WWW in Form der Verzeichnisdienste eine breite Anwendung. Um diese Verzeichnisdienste für das Semantic Web nutzbar zu machen, müssen die Kategorisierungsschemata und Verzeichniseinträge in maschinenlesbarer Form zur Verfügung gestellt werden. Diese Anforderung wird zum Beispiel vom im Rahmen des Projektes *Open Directory Project* erstellten Verzeichnis erfüllt [31]. Allerdings basieren die dort verwendeten Verzeichnisauszüge in RDF (so genannte *RDF Dumps*) auf einer nicht-standardisierten frühen Arbeitsversion und müssen vor einer Weiterverarbeitung bereinigt werden. Dazu können Skripten von der Homepage des Projektes *Digital Libraries Project, Database Group, Stanford University* kostenlos heruntergeladen werden.

Das zweite Konzept wird im WWW im Wesentlichen vom menschlichen Benutzer realisiert, der die gefundenen Informationen selber in seine Sprache übersetzen oder eine ihm fremde Sprache (z. B. ein Fachjargon) zuerst erlernen muss. Dabei stehen ihm zum Teil elektronische Wörterbücher und Fachglossare unterstützend zur Verfügung. Zur Entwicklung der für die Automatisierung dieser Aufgaben im Semantic Web nötigen Ontologien bietet das W3C mit OWL (*Web Ontology Language*) eine standardisierte Beschreibungssprache an [30].

## Bedingungen für Software-Agenten

Wie erwähnt formuliert das W3C mit den für das Semantic Web spezifischen Empfehlungen stillschweigend Bedingungen für Software-Agenten. Eine erste Bedingung leitet sich aus der Tatsache ab, dass alle Technologien des Semantic Web auf *Extensible Markup Language (XML)* basierende (Austausch-) Syntaxen zur Verfügung stellen: Ein Software-Agent muss XML lesen und schreiben (bzw. „sprechen“) können.

Die Beschreibung von Informationsquellen mit *Resource Description Framework (RDF)* unterstellt, dass das Wissen eines Software-Agenten (über diese Informationsquellen) als eine Menge von *Property Statements* strukturiert ist, das heisst als binäre Relationen oder Tripel der allgemeinen Form (*Prädikat Subjekt Objekt*). Obwohl eine solche Strukturierung nicht zwingend ist, wird dadurch die Konzeption eines Kommunikationsprotokolls für Software-Agenten wesentlich vereinfacht [19].

Kennzeichnend für einen Software-Agenten des Semantic Web ist, dass er Beschreibungen in RDF oder OWL auf eine Logik erster Ordnung abbilden kann. Eine solche Abbildung ermöglicht es ihm, die Wahrheitswerte (wahr oder falsch) von Ausdrücken zu ermitteln, die nach bestimmten Regeln von Aussagen mit bekannten Wahrheitswerten abgeleitet sind und so die Ausdrücke zu „verstehen“. Diese Eigenschaft ist unter anderem für die Verifizierung von Beweisen nötig, welche zusammen mit digitalen Unterschriften die Grundlage für das „Web of Trust“ – die (vorläufig) letzte Ausbaustufe des Semantic Web – bilden. Die Definition und Instanziierung von Ontologien in OWL – namentlich in OWL Lite und OWL DL – legt die erwähnte Logik erster Ordnung auf die diesen Sprachen zugrunde liegenden Beschreibungslogiken  $\mathcal{SHIQ}$  und  $\mathcal{SHOQ}(D)$  fest [23, 32].

Die Technologien des Semantic Web gründen auf Konzepten aus dem als *Wissensrepräsentation und Inferenz* bezeichneten Teilgebiet der Künstlichen Intelligenz. Diese Grundlage erklärt, in welchem Sinne die Software-Agenten des Semantic Web lernfähig sind.

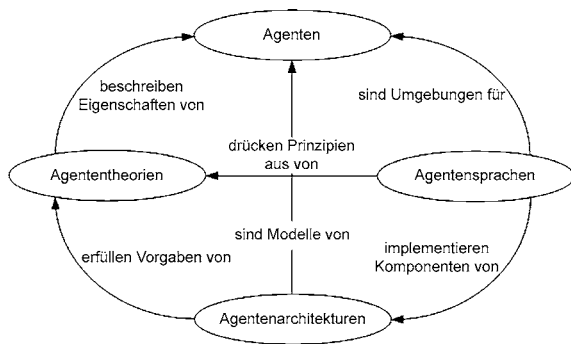
Versteht man Lernen als den Erwerb von (zusätzlichem) Wissen auf der Grundlage von gegebenen konzeptuellen Strukturen, dann sind die Software-Agenten des Semantic Web uneingeschränkt lernfähig. Versteht man dagegen Lernen als das (Weiter-) Entwickeln dieser Strukturen, dann lernen die Software-Agenten des Semantic Web ausschliesslich in Verbindung mit der *Evolution* von Ontologien [26].

Bei der Erschliessung von neuem Wissen werden ähnliche Konzepte oft gleichzeitig von mehreren Forschungsgruppen definiert oder von einer einzelnen Gruppe zu verschiedenen Zeitpunkten. Im ungünstigsten Fall wird so ein und dasselbe Konzept – unter demselben oder einem anderen Namen – in mehreren Ontologien mit unterschiedlicher räumlicher oder zeitlicher Gültigkeit beschrieben. Ein Beispiel ist der Begriff des Agenten, der sich mit der Entwicklung des Fachgebietes der Künstlichen Intelligenz gewandelt hat [1]. Um Informationen, die dieses Konzept benutzen, miteinander kombinieren zu können, müssen die verschiedenen Beschreibungen zueinander in Beziehung gesetzt werden [26]. Mit dem ausdrücklichen Festlegen der geltenden Beziehungen vollziehen die Software-Agenten des Semantic Web die mit der Evolution verbundenen Lernschritte. Die Evolution von Ontologien steht im scharfen Kontrast zum *induktiven* Lernen, das heisst zum Lernen anhand von Beispielen, welches das Lernen mit neuronalen Netzen auszeichnet.

## Beschreibung der Software-Agenten

Die Software-Agenten des Semantic Web lassen sich anhand der gebräuchlichen Agententheorien, -architekturen und -sprachen beschreiben. Die Auswahl der für die Beschreibung verwendeten Agententheorien, -architekturen und -sprachen erfolgt anhand der Übersicht von Wooldridge und Jennings [42]. Derselben Arbeit sind die Erklärungen für die drei Grundbegriffe entnommen.

- *Agententheorien* versuchen, Agenten begrifflich zu fassen, die Eigenschaften von Agenten zu verstehen, festzulegen und formal darzustellen. Aus Sicht des



**Abb. 4 Beziehungen zwischen Agenten, Agententheorien, -architekturen und -sprachen**

Software-Entwicklungsprozesses sind Agententheorien im Wesentlichen Spezifikationen.

- **Agentenarchitekturen** versuchen die in den Agententheorien beschriebenen Eigenschaften bei der Gestaltung von Rechnersystemen umzusetzen. Sie befassen sich mit den dafür geeigneten Strukturen der Hard- und Software sowie mit der Abgrenzung von Komponenten. Aus Sicht des Software-Entwicklungsprozesses sind Agentenarchitekturen Modelle von Agenten. Sie geben den Schritt von der Spezifikation zur Implementierung wieder.
- **Agentensprachen** (abzugrenzen von den oben behandelten *Agentenkommunikationssprachen*!) sind Programmiersprachen, welche die verschiedenen in den Theorien vorgeschlagenen Prinzipien ausdrücken. Sie befassen sich mit dem Vorgehen bei der Programmierung von Agenten, mit der Ermittlung der dazu nötigen Primitiven und mit der effektiven Kompilierung und Ausführung von Agentenprogrammen. Aus Sicht des Software-Entwicklungsprozesses stellen Agentensprachen die Entwicklungs- und Testumgebungen für Agenten zur Verfügung.

Die Beziehungen zwischen Agenten, Agententheorien, -architekturen und -sprachen werden in der Abb. 4 verdeutlicht.

### Agententheorien

Die theoretischen Grundlagen für die Software-Agenten des Semantic Web liefern die Beschreibung von Agenten als rational und vorsätzlich handelnde Systeme, die Semantik der möglichen Welten und die Sprechakttheorie.

*Rational handelnde Agenten (Rational Agents)* handeln insofern „vernünftig“, als ihre Handlungen vollständig im ihnen verfügbaren Wissen und dem mit den Methoden der (klassischen) Logik daraus ableitbaren Wissen gründen. Eng verwandt mit dem Begriff des rational handelnden Agenten ist die Beschreibung von Agenten als *vorsätzlich handelnde*

*Systeme (Intentional Systems)*. Dabei wird den Agenten eine Anzahl von (inneren) Einstellungen oder Haltungen (*Attitudes*) – wie zum Beispiel Wissen – zugeschrieben, die ihr Verhalten erklären.

Dass die Software-Agenten des Semantic Web Wissen haben und Handlungen ausführen ist unbestritten. Hawke macht zudem deutlich, dass die Handlungssteuerung mit dem Wissen erfolgt, das die Agenten erwerben oder erschliessen [19]. Weil die Erschliessung von neuem Wissen im Semantic Web bisher ausschliesslich mit den Methoden der klassischen Logik erfolgt (sowohl die Beschreibungslogik als auch die Hornlogik sind Teilmengen der Prädikatenlogik erster Stufe [18]), sind die Software-Agenten des Semantic Web im oben definierten Sinne rational handelnde Agenten.

Abgesehen vom Wissen machen die Entwickler des Semantic Web keine Angaben zu den Haltungen von Software-Agenten. Es ist aber anzunehmen, dass zumindest die von Genesereth und Ketchpel erwähnten – nämlich Aufrichtigkeit, Autonomie und Verpflichtung – vorausgesetzt werden [16]. Im zunehmend für die wirtschaftliche Leistungserstellung genutzten Semantic Web ist auch der *Warenkorb* als Bestand an digitalen oder digital repräsentierten Gütern und Geld eine wichtige Voraussetzung [36, 37].

Nach der *Semantik der möglichen Welten (Possible Worlds Semantics)* können die Überzeugungen eines Agenten als eine Menge von möglichen Welten aufgefasst werden. Jede dieser Welten stellt einen möglichen Sachverhalt für das (begrenzte) Wissen eines Agenten dar. Hintikka prägte den Begriff der *epistemischen Alternativen*, um die möglichen Welten zu beschreiben [21]. Etwas, das in allen epistemischen Alternativen eines Agenten wahr ist, wird als seine Überzeugung bezeichnet.

Epistemische Logiken, mit denen Modelle wie das der möglichen Welten formal beschrieben werden, werden normalerweise als normale Modallogiken mit der von Kripke entwickelten Semantik formuliert [28]. Eine normale Modallogik ist im wesentlichen klassische Aussagenlogik, die um die zwei Operatoren  $\Box$  (notwendigerweise) und  $\Diamond$  (möglicherweise) erweitert worden ist. Die Formel  $\Box\varphi$  wird als „notwendigerweise  $\varphi$ “ gelesen und  $\Diamond\varphi$  als „möglicherweise  $\varphi$ “.

Die Semantik der modalen Junktoren  $\Box$  und  $\Diamond$  wird durch die Einführung einer *Zugänglich-*

*keitsrelation (Accessibility Relation)* in die für die Sprache verwendeten Modelle festgelegt. Diese Relation definiert, welche Welten als von jeder anderen Welt aus erreichbar betrachtet werden. Die Formel  $\Box\varphi$  ist dann wahr, wenn  $\varphi$  in *jeder* von der aktuellen Welt aus erreichbaren Welt wahr ist.  $\Diamond\varphi$  ist dann wahr, wenn  $\varphi$  *mindestens* in einer von der aktuellen Welt aus erreichbaren Welt wahr ist.

Bei der Verwendung dieser Logik als epistemische Logik wird  $\Box\varphi$  als „es ist bekannt, dass  $\varphi$ “ gelesen. Die Welten im Modell werden als epistemische Alternativen gedeutet, und die Zugänglichkeitsrelation definiert, welches die Alternativen für eine gegebene Welt sind. Zwei grundlegende Eigenschaften der normalen Modallogik begründen bei der epistemischen Logik das Problem der *logischen Allwissenheit (Logical Omniscience)*: Sie besagen, dass ein Agent alle gültigen Formeln und alle logischen Konsequenzen seiner Überzeugungen kennt. Zur Beschreibung des Wissens eines Agenten mit beschränkten Ressourcen scheint das Modell der möglichen Welten deshalb ungeeignet zu sein.

Eine Lösung des Problems der logischen Allwissenheit besteht darin, die Menge der möglichen Welten auf die Menge der Welten mit einer konkreten Interpretation zu beschränken. Einen ähnlichen Ansatz verfolgt das Semantic Web mit den Ontologien: Eine von einem Agenten benutzte Ontologie ist eine Abstraktion (oder Theorie) der möglichen Welten des Agenten, der aktuelle Zustand seiner Wissensbasis die aktuelle Welt, und die Menge der von diesem Zustand aus erreichbaren Zustände ist die Menge der möglichen Welten des Agenten. Im Falle einer in OWL definierten Ontologie sind diese Welten in Form eines formalen Modelles semantisch präzise beschrieben [32].  $\Box\varphi$  bedeutet dann Konsistenz der aktuellen Wissensbasis des Agenten *in Bezug auf die verwendete Ontologie*: Es gibt eine Interpretation, die gleichzeitig ein Modell der aktuellen Wissensbasis (der aktuellen Welt) und der verwendeten Ontologie (als Abstraktion der möglichen Welten) ist. Umgekehrt gibt es für jede mögliche Wissensbasis, die in Bezug auf die verwendete Ontologie konsistent ist, eine solche Interpretation. Zur Prüfung auf Konsistenz mit einer in OWL definierten Ontologie können Algorithmen verwendet werden, das heisst, die Prüfung ist automatisierbar [23].

Die von John L. Austin und John R. Searle begründete *Sprechaktttheorie* wurde bereits im Abschnitt *Eine prototypische Agentenkommunikationssprache* erwähnt. Für eine ausführliche Diskussion sei auf Searle verwiesen [38].

## Agentenarchitektur

Die Architektur der Software-Agenten des Semantic Web folgt dem klassischen Ansatz der *deliberativen* Architektur. Dabei werden Agenten als eine besondere Art von wissensbasierten Systemen konstruiert. Wooldridge und Jennings definieren einen Agenten mit einer deliberativen Architektur, als einen Agenten mit einem explizit repräsentierten, symbolischen Modell der Welt, welcher seine Entscheidungen (z. B. welche Handlungen auszuführen sind) mit Hilfe von logischen Schlussfolgerungen auf der Grundlage der Verarbeitung von Symbolen trifft [42].

Im Semantic Web werden symbolische Modelle der Welt als (instanzierte) Ontologien repräsentiert. Weil diese standardmässig in OWL definiert werden, sind die Schlussfolgerungen zunächst auf jene beschränkt, die in den zugrunde liegenden Beschreibungslogiken zulässig sind. Mit OWL<sup>-</sup>, *Description Logic Programs (DLP)* und *Semantic Web Rule Language (SWRL)* wird aber eine Verbindung dieser Beschreibungslogiken mit Hornlogik gesucht [10, 18, 22]. Dadurch soll das Repertoire der von den Software-Agenten des Semantic Web unterstützten Folgerungen reichhaltiger gestaltet werden. Hornlogik liegt auch den ebenfalls für das Semantic Web vorgesehenen Sprachen RuleML und *RuleML Rules Lite*, welche mit RDF und OWL DL kompatibel ist, zugrunde [6, 7].

## Agentensprache

Ein Vorteil des Ansatzes von DARPA KSE, dass Agenten aufgrund ihrer Fähigkeit, korrekt in einer Sprache wie ACL zu kommunizieren, definiert werden, ist, dass dadurch die Schnittstelle eines Agenten von seiner Implementierung entkoppelt wird [16]. Falls die Entwickler des Semantic Web mit demselben Ansatz arbeiten, können die Software-Agenten des Semantic Web in einer beliebigen Programmiersprache (bzw. im Sinne der oben eingeführten Begrifflichkeit Agentensprache) implementiert werden, solange sie die Kommunikation in der gemeinsamen Sprache unterstützen. Der Agent des *Knowledge Systems Laboratory* der





## Theorien, Architektur und Sprache der Software-Agenten des Semantic Web

	Beschreibung	Semantic Web
Agententheorie	Agenten als rational und vorsätzlich handelnde Systeme	Agenten handeln aufgrund des erworbenen und mit klassischer Logik erschlossenen Wissens
	Semantik der möglichen Welten	Ontologien als Abstraktionen der möglichen Welten
	Sprechakttheorie	RuleML
Agentenarchitektur	Deliberative Architektur	Agenten als (besondere) wissensbasierte Systeme
Agentensprache	Programmiersprache einer konkreten Implementierung	Entkoppelung der Schnittstelle (Kommunikation) von der Implementierung

Stanford University, welcher Fragen in OWL-QL beantwortet, ist zum Beispiel in der Programmiersprache Java implementiert [15].

Die dem Semantic Web zugrunde liegenden Agententheorien, -architekturen und -sprachen und ihre Umsetzung im Semantic Web sind in Tabelle 1 zusammengefasst.

### Anwendungen von Software-Agenten

Vom W3C werden Jema, Zakim und der Wein-Agent der Stanford University als Anwendungen von Software-Agenten des Semantic Web nachgewiesen. Listen mit auch ausserhalb vom W3C entwickelten Anwendungen finden sich unter den Webangeboten des EU-Projektes *Semantic Web Advanced Development for Europe (SWAD-Europe)* und des Programmes *DARPA Agent Markup Language (DAML)* [9, 41].

*Jema*, der *Jena Meeting Assistant*, ist eine RDF-Anwendung, welche die Abläufe in den Arbeitsgruppen des W3C unterstützt [41]. *Jena* ist eine als Open Source verfügbare Umgebung zur Entwicklung von Anwendungen für das Semantic Web in Java [25]. Mit Hilfe von Punkte- und Aufgabenlisten unterstützt Jema das Anlegen und Nachführen von Tagesordnungspunkten und das Verfolgen des Arbeitsfortschritts. Jema bindet E-Mail-Listen und *Internet Relay Chat (IRC)* ein und kann während einer wöchentlichen Telefonkonferenz herangezogen werden, um bei der Terminplanung mitzuhelfen.

Der IRC-Telefonkonferenz-Agent *Zakim* unterstützt Konferenzen in einem zum Telefon parallelen IRC-Kanal zur Echtzeit [27]. Im konkreten Fall liest er die Tagesordnung aus einer RDF/XML-Datei. Anders als Jema kann Zakim über eine Brücke zum Telefon Informationen einbinden, die sonst nur dem

Betreiber zur Verfügung stehen. Auf diese Weise gibt er bei Bedarf darüber Auskunft, wer alles anwesend ist, meldet es, wenn neue Teilnehmer dazustossen oder ausscheiden, reiht Wortmeldungen ein, ruft die Teilnehmer auf, wenn die Reihe an ihnen ist, und führt die Tagesordnung. Das Protokoll wird im IRC-Kanal geführt und unrichtige Einträge sofort berichtigt. Zur Veröffentlichung des Protokolls wird im IRC-Kanal ein zweiter Agent aufgerufen. Zakim kann mehrere Konferenzen gleichzeitig unterstützen. Er hat eine WWW-Schnittstelle, die den jeweiligen Zustand der Brücke auf dynamisch erzeugten Seiten anzeigt.

Der *Wein-Agent* der Stanford University, Knowledge Systems Laboratory, tritt als Weinführer auf. Unter gegebenen Randbedingungen, wie zum Beispiel das servierte Essen, empfiehlt er Weine, findet Informationen zu einem bestimmten Wein oder einer bestimmten Weinklasse, sucht nach passendem Zubehör zu einem Wein, wie zum Beispiel eine besondere Art von Gläsern. Der Wein-Agent verbindet über eine Schnittstelle eine Problemlösungskomponente (*Reasoner*) mit einer in OWL definierten Ontologie [15]. Der Arbeitsgang des Agenten kann in drei Schritten beschrieben werden: in der Ontologie nachschlagen, Suchanfragen bearbeiten und Ergebnisse ausgeben. Der Wein-Agent verwendet OWL-QL als äussere Sprache im Sinne der im Abschnitt *Software-Agenten* eingeführten Agentenkommunikationssprache und hat eine WWW-Schnittstelle, an welcher der Benutzer in KIF Suchanfragen stellen kann.

### Danksagung

Ein Teil der Arbeiten, die diesem Artikel zugrunde liegen, wurde von der Europäischen Kommission und dem Schweizerischen Staatssekretariat für

Bildung und Forschung durch das im sechsten Rahmenprogramm unter dem Schwerpunkt *Information Society Technologies* eingesetzte REWERSE *Network of Excellence* (Nr. 506779) unterstützt.

Das *Semantic Web* erweitert das WWW dadurch, dass zu den Informationen Beschreibungen hinzugefügt werden. Diese Beschreibungen sind in einer Sprache mit einer formalen Semantik verfasst und ermöglichen es besonders dafür ausgelegten Software-Agenten, die beschriebenen Informationen zu „verstehen“. Dieses Verständnis erleichtert die Zusammenarbeit der Benutzer mit den Software-Agenten bei der Lösung von Problemen.

Die *Software-Agenten* des *Semantic Web* führen – in teilweiser Autonomie – für ihre Benutzer Aufträge aus. Sie haben Wissen, sind zu gewissen Folgerungen (oder *Inferenzen*) befähigt und kommunizieren in einer gemeinsamen Agentenkommunikationssprache. Diese gemeinsame Sprache (oder dieses Portfolio von Sprachen) ist Gegenstand der laufenden Forschung.

Eine *Ontologie* ist eine formale Beschreibung eines sprachlich erschlossenen Gegenstandsbereiches. Sie besteht aus einem Klassifizierungssystem (oder einer *Taxonomie*) und Axiomen, welche die Interpretation und den korrekten Gebrauch der Klassenbegriffe regeln. In Verbindung mit der Beschreibungslogik wird als *Ontologie* eine zur Beschreibung der Welt definierte Terminologie bezeichnet und als *instanzierte Ontologie* eine Ontologie mit einer Beschreibung der Welt, das heisst mit einer Wissensbasis.

Die allgemeine Sprachtheorie unterscheidet drei Ebenen der *Kommunikation*. Die Ebene der *Syntaktik* befasst sich mit der korrekten Übertragung der Zeichen und mit ihrer formalen Beziehung zueinander. Auf der Ebene der *Semantik* stehen die Bedeutung der Zeichen und die Inhalte der Nachrichten im Vordergrund. Die Ebene der *Pragmatik* befasst sich mit der Wirkung, die der Sender mit seiner Nachricht beim Empfänger erzielen möchte. Diesen drei Ebenen entsprechen die drei Teile der Agentenkommunikationssprache des *Semantic Web*.

## Literatur

1. AgentLink: Agent Technology Roadmap: Overview and Consultation Report. European Commission's Sixth Framework Programme (FP6), Information Society Technologies (IST), Agent-Based Computing 2004 (<http://www.agentlink.org/roadmap/roadmapreport.pdf>)
2. Berners-Lee, T.: A roadmap to the Semantic Web. 1998 (<http://www.w3.org/DesignIssues/Semantic.html>)
3. Berners-Lee, T.: Webizing existing systems (rev. ed.). 2001, May 22 (<http://www.w3.org/DesignIssues/Webize.html>)
4. Berners-Lee, T.: Keynote. The Thirteenth International World Wide Web Conference (WWW2004). (<http://www.w3.org/2004/Talks/0519-tbl-keynote/>)
5. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, May 2001
6. Boley, H., Tabet, S.: RuleML Rules Lite. Concrete Syntax. 2003, September 30 (<http://www.ruleml.org/submission/ruleslite/concretesyntax.html>)
7. Boley, H., Tabet, S.: The Rule Markup Initiative (rev. ed.). 2005, January 18 (<http://www.ruleml.org/>)
8. Connolly, D.: Knowledge Interchange Format (KIF) as an RDF Schema (rev. ed.). 2000, August 14 (<http://www.w3.org/2000/07/hs78/KIF.html>)
9. DAML: DARPA Agent Markup Language. Applications (Version 1.22). 2003 (<http://www.daml.org/applications/>)
10. de Bruijn, J., Polleres, A., Lara, R., Fensel, D.: OWL<sup>-</sup>. Working Draft D20.1v0.2. 2004, October 12 (<http://www.wsmo.org/2004/d20/d20.1/v0.2/>)
11. Fikes, R., Hayes, P., Horrocks, I.: OWL-QL – A Language for Deductive Query Answering on the Semantic Web. 2003 ([http://ksl-web.stanford.edu/KSL\\_Abstacts/KSL-03-14.html](http://ksl-web.stanford.edu/KSL_Abstacts/KSL-03-14.html))
12. Finin, T., Fritzson, R., McKay, D.: A Language and Protocol to Support Intelligent Agent Interoperability. In: Proceedings of the CE & CALS Washington '92 Conference. 1992, June (<http://www.cs.umbc.edu/kqml/papers/>)
13. Finin, T., Fritzson, R., McKay, D., McEntire R.: KQML – A language and protocol for knowledge and information exchange. In: Proceedings of the 13th International Workshop on Distributed Artificial Intelligence, Seattle, WA, (126–136). 1994, July (<http://www.cs.umbc.edu/kqml/papers/kbks.pdf>)
14. Finin, T., Fritzson, R., McKay, D., McEntire, R.: KQML as an Agent Communication Language. In: N. Adam, B. Bhargava, Y. Yesha (Eds.): Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94) (pp. 456–463). Gaithersburg, MD, USA: ACM Press 1994, November (<http://www.cs.umbc.edu/kqml/papers/>)
15. Frank, G., Jenkins, J., Fikes, R.: JTP: An Object-Oriented Modular Reasoning System (rev. ed.). 2004, July 8 (<http://ksl.stanford.edu/software/JTP/>)
16. Genesereth, M.R., Ketchpel, S.P.: Software Agents. Communications of the ACM 37 (7), 48–53 (1994)
17. Genesereth, M.R.: Knowledge Interchange Format. Draft proposed American National Standard (dpANS). NCITS.T2/98-004. 1998 (<http://logic.stanford.edu/kif/dpans.html>)
18. Grosz, B.N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logic. In: Proceedings of the Twelfth International World Wide Web Conference (WWW2003), Budapest, Hungary (pp. 48–57). New York: ACM Press 2003 (<http://www.cs.man.ac.uk/~horrocks/Publications/download/2003/p117-grosz.pdf>)
19. Hawke, S.: Semantic Web Architecture. 2001, May 14 (<http://www.w3.org/2001/05/14/swarch/>)
20. Hendler, J.: Agents and the Semantic Web (Preprint). IEEE Intelligent Systems Journal 16 (2), 30–37 (2001) (<http://www.cs.umd.edu/users/hendler/AgentWeb.html>)
21. Hintikka, J.: Knowledge and Belief. Ithaca, NY: Cornell University Press 1962
22. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., Dean, M.: SWRL: A Semantic Web Rule Language. Combining OWL and RuleML (Version 0.5). 2003, November 19 (<http://www.daml.org/2003/11/swrl/>)
23. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. Journal of Web Semantics 1 (1), 7–26 (2003)
24. Jacobs, I., Walsh, N.: Architecture of the World Wide Web, Volume One. W3C Recommendation 15 December 2004. (<http://www.w3.org/TR/2004/REC-webarch-20041215/>)
25. Jena: A Semantic Web Framework for Java. o.D. (<http://jena.sourceforge.net/>)
26. Koivunen, M.-R., Miller, E.: W3C Semantic Web Activity. In: Proceedings of the Semantic Web Kick-off Seminar in Finland (pp. 27–44). 2001, November (<http://www.cs.helsinki.fi/u/eahyvone/stes/semanticweb/kick-off/proceedings.pdf>)
27. Kotok, A., Swick R.: The Zakim IRC Teleconference Agent. World Wide Web Consortium 2001 (<http://www.w3.org/2001/12/zakim-irc-bot.html>)

28. Kripke, S.: Semantical analysis of modal logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 9, 67–96 (1963)
29. McCool, R.: OWL-QL Syntax Overview. 2002 [<http://www.ksl.stanford.edu/projects/owl-ql/syntax.shtml>]
30. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language: Overview. W3C Recommendation 10 February 2004. [<http://www.w3.org/TR/2004/REC-owl-features-20040210/>]
31. Open Directory Project. 1998–2005 [<http://dmoz.org/>]
32. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL Web Ontology Language: Semantics and Abstract Syntax. W3C Recommendation 10 February 2004. [<http://www.w3.org/TR/2004/REC-owl-semantics-20040210/>]
33. Peer, J.: A POP-based Replanning Agent for Automatic Web Service Composition. In: A. Gómez-Pérez & J. Euzenat (Eds.): *Proceedings of the Second European Semantic Web Conference (ESWC 2005)*, Heraklion, Greece. The Semantic Web: Research and Applications. *Lecture Notes in Computer Science* 3532 (pp. 47–61). Berlin: Springer 2005
34. Reichwald, R.: Informationsmanagement. In: M. Bitz, K. Dellmann, M. Domsch, F.W. Wagner (Hrsg.): *Vahlens Kompendium der Betriebswirtschaftslehre* (4. Aufl., Band 2, S. 221–288). München: Vahlen 1999
35. REVERSE – Reasoning on the Web with Rules and Semantics (rev. ed.). 2004, December 21 [<http://reverse.net>]
36. Schmid, B.F.: Elektronische Märkte – Merkmale, Organisation und Potentiale. In: M. Sauter, & A. Hermanns (Hrsg.): *Management-Handbuch Electronic Commerce* (S. 31–48). München: Franz Vahlen 1999
37. Schmid, B.F.: Elektronische Märkte. In: R. Weiber (Hrsg.): *Handbuch Electronic Business: Informationstechnologien – Electronic Commerce – Geschäftsprozesse* (S. 181–207). Wiesbaden: Gabler 2000 [<http://www.mediamanagement.org/modules/pub/view.php/businessmedia-14>]
38. Searle, J.R.: *Speech Acts. An Essay in the Philosophy of Language* (Seventh reprint of 1978). New York: Cambridge University Press 1969
39. Shoham, Y.: Agent-oriented programming. *Artificial Intelligence* 60, 51–92 (1993)
40. Stanoevska, K.: The Concept of Knowledge Media: The Past and Future. In: R. Grütter (Ed.): *Knowledge Media in Healthcare: Opportunities and Challenges*. Hershey: Idea Group Publishing 2002
41. SWAD-Europe: Semantic web applications - analysis and selection. Appendix B – Application Survey (Deliverable 12.1.1). European Commission's Seventh Framework Programme (FP7), Information Society Technologies (IST) 2001 [<http://www.w3.org/2001/sw/Europe/reports/pdf/12.1.1.appendixA.pdf>]
42. Wooldridge, M., Jennings, N.R.: Intelligent Agents: Theory and Practice. *Knowledge Engineering Review* 10 (2), 115-152 (1995) [<http://www.csc.liv.ac.uk/~mjw/pubs/ker95.pdf>]